

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-096818

(43)Date of publication of application : 30.03.1992

(51)Int.Cl.

G06F 3/06  
G06F 3/06  
G11B 17/22  
G11B 19/02  
G11B 20/00  
G11B 20/12  
G11B 20/18  
G11B 20/18  
G11B 27/10

(21)Application number : 02-214261

(71)Applicant : HITACHI LTD

(22)Date of filing : 15.08.1990

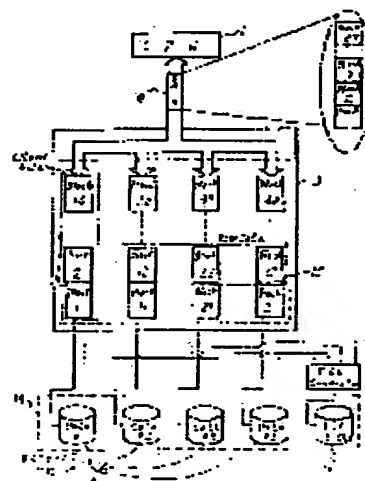
(72)Inventor : TSUNODA HITOSHI  
KAMO YOSHIHISA

## (54) COLLECTIVE DISK DEVICE

### (57)Abstract:

**PURPOSE:** To increase the transfer speed to improve the processing efficiency by writing matrix data corresponding to a disk drive in parallel and reading out data from the disk drive with one block as the unit.

**CONSTITUTION:** Write data transferred from a CPU 1 at a high speed in serial is distributed as matrix data 10 corresponding to an ECC disk group 14 in a data buffer 3. Data 9 and data 10 consists of many blocks. After data 9 is expanded to data 10, it is collectively written on data disks 4 in a group 14 from the buffer 3. At the time of write to disks 4, blocks are transferred to an ECC generator 6 also, and ECC is generated by blocks and is recorded on an ECC disk 5. At the time of read, only disks 4 where data is stored are accessed to read out data. Thus, the speed of transfer from the CPU is increased to perform the processing at a high speed.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

平4-96818

⑤ Int. Cl.<sup>5</sup>

G 06 F 3/06

G 11 B 17/22

19/02

20/00

20/12

20/18

27/10

識別記号

3 0 1 R

3 0 1 N

3 0 5 C

F

D

1 0 1

1 0 2

D

庁内整理番号

7232-5B

7232-5B

7232-5B

7719-5D

7627-5D

9197-5D

9074-5D

9074-5D

9074-5D

8224-5D

⑬ 公開 平成4年(1992)3月30日

審査請求 未請求 請求項の数 12 (全15頁)

⑭ 発明の名称 集合ディスク装置

⑮ 特 願 平2-214261

⑯ 出 願 平2(1990)8月15日

⑰ 発 明 者 角 田 仁 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑰ 発 明 者 加 茂 善 久 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑱ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑲ 代 理 人 弁理士 小川 勝男 外1名

明 細 書

# 1. 発明の名称

集合ディスク装置

# 2. 特許請求の範囲

## 1. 上位装置からのデータの入出力要求に対する、

当該データを格納するディスクドライブと、該ディスクドライブを管理する制御装置からなるシステムにおいて、上記上位装置からシリアルに転送されてきたデータを分割し、さらにそれらのデータからエラー検出、ECCを生成し、分割したデータおよびECCを別々のディスクドライブに並列で一度に格納することによりグループを作成し、読出しの場合は個々のデータのみへのアクセスを行うことを特徴とする集合ディスク装置。

## 2. 前記集合ディスク装置において、上位装置から転送されてくる書き込みデータの大きさが一定である場合、その書き込みデータをあらかじめ決めておいた容量に分割し複数の領域に分けておき、その各々の領域を一端、半導体メモリに格

納し、そこで、一度に並列に格納するディスクドライブの台数と同じ数に分割することにより、マトリックス状のデータに変換することの特徴とする特許請求範囲第1項記載の集合ディスク装置。

## 3. 前記集合ディスク装置において、データ格納用ディスクドライブとECC格納用ディスクドライブにデータ及びECCを格納する際、ECCの格納の終了を待たずに次のデータの格納を開始してしまうことを特徴とする特許請求範囲第1項記載の集合ディスク装置。

## 4. 前記集合ディスク装置において、少なくともデータおよびECCをパラレルに書き込む時と、ディスクドライブに障害が発生し、その中のデータを復元するため、残りのデータとECCのデータを読み出す時に、当該データ格納用ディスクドライブとECC格納用ディスクドライブの回転を同期させることを特徴とする特許請求範囲第1項記載の集合ディスク装置。

## 5. 前記集合ディスク装置において各データ格納

用ディスクドライブにバッファを設け、ディスクドライブとバッファの双方にデータディスクに格納されているデータのカatalogを格納しておくことにより、データ格納用ディスクドライブからデータを読み込む場合、そのアドレスはバッファを見ることにより行なうことを特徴とする特許請求範囲第1項記載の集合ディスク装置。

6. 前記集合ディスク装置において上位装置からシリアルに転送されてきたデータを、マトリックス状のデータに変換しパラレルに格納するディスクドライブ数に分割した際、当該ディスクドライブに格納されるデータの長さが揃わない場合不足している分だけ、ある特殊なデータを埋め込み、長さを揃えることを特徴とする特許請求範囲第1項記載の集合ディスク装置。

7. 前記集合ディスク装置において上位装置からシリアルに転送されてきたデータを、パラレルに格納するディスクドライブに、ある単位で順に振り分け、マトリックス状のデータを作成し

り構成されるグループにおいて、一つのディスクドライブのシリンダへの書き込みが終了した時点でグループ内の他のディスクドライブへの書き込みも終了として、グループ内の全てのディスクドライブにおいてシリンダを移動することを特徴とする特許請求範囲第1項記載の集合ディスク装置。

11. 前記集合ディスク装置において、複数のディスクドライブにより構成される一括書き込みを行う単位であるグループにおいて、そのグループ内の個々のディスクドライブの中で、少なくとも一台でも使用されている場合、“ON”とするグループの使用を示すフラグを管理テーブルとして、集合ディスク装置を制御する部分に設け、一括書き込みを行う場合そのフラグを見ることにより書き込み可能かを判定することを特徴とする特許請求範囲第1項記載の集合ディスク装置。

12. 前記集合ディスク装置において、複数のディスクドライブにより構成される一括書き込みを

データがそろい次第パラレルに転送することを特徴とする特許請求範囲第1項記載の集合ディスク装置。

8. 前記集合ディスク装置においてを当該データ格納用ディスクドライブにより構成されるグループを論理的なボリュームとして扱うことを特徴とする特許請求範囲第1項記載の集合ディスク装置。

9. 前記集合ディスク装置において、一括書き込みを行っている際当該ディスクドライブ内の当該トラックが書き込み不能の場合、そのトラックへの書き込みは行わず、又、該ディスク上の予備のトラックへヘッドを移動して書き込む動作を行わずに、そのトラックをとばして、同一シリンダ内の他のトラックへ、ヘッドの電気的な切り替えにより、次のトラックへ書き込むことを特徴とする特許請求範囲第1項記載の集合ディスク装置。

10. 前記集合ディスク装置において、一括書き込みを行っている際、当該ディスクドライブによ

り行う単位であるグループにおいて、そのグループ内の個々のディスクドライブについて使用されている場合、“ON”とする使用中フラグと、そのグループ内のディスクドライブが少なくとも一つでも使用されている場合“ON”とする使用中フラグを、階層的な管理テーブルとして集合ディスク装置を制御する部分に設けることを特徴とする特許請求範囲第1項記載の集合ディスク装置。

### 3. 発明の詳細な説明

#### (産業上の利用分野)

本発明はコンピュータシステムに係り、特に高性能な入出力動作を可能とするディスクファイルシステムに関する。

#### (従来の技術)

現在のコンピュータシステムにおいては、CPU等の上位側が必要とするデータは2次記憶装置に格納され、CPUが必要とする時に応じて2次記憶装置に対してデータの書き込み、読みだしを行っている。この2次記憶装置としては一般に不揮発

な記憶媒体が使用され、代表的なものとして磁気ディスク、光ディスクなどが使用されている。特許、学術文献等のデータベースなどでは、通常のユーザが使用する場合は2次記憶装置から読み出すのみで、ユーザが直接データを書換えることはない。データの変更、追加を行う場合は、データベースを管理する者が一度にまとめて追加という形で行う。

従来、多数の比較的容量の小さな磁気ディスクを用意し、CPUから転送されてくるデータを分割し、複数の磁気ディスクに同時に格納し、読み出す場合は逆に各々の磁気ディスクから同時に読み込んできてCPUへ転送するパラレル処理を行うという技術がある。このようなシステムについてはストリッジ コンセプト社等の企業から製品発表等されている。また、情報処理学会第39回全国大会において桜井 紀彦らが“アレイ制御ディスクの構成方式に関する一考察”においてデータを並列に処理したり、通常のようにシリアルなデータとして処理したりを一つのシステム内にお

いてダイナミックに切り替えて行う方式について報告している。

〔発明が解決しようとする課題〕

特許、学術文献等のデータベースなどでは、通常のユーザが使用する場合は2次記憶装置から個々のデータを読み出すのみで、ユーザが直接データを書換えるようなことはない。データの変更、追加を行う場合は、データベースを管理する者が一度にまとめて追加という形で行う。この追加作業はCPUから送られてくる大量のシリアルなデータを、それを構成する個々のアクセス単位であるブロックについて細かく、しかも順次行われるため、書込み時間が大量にかかる。また、上記したようにシリアルに転送されてくるデータ内の個々のデータについて行うことから、磁気ディスクの処理時間により上位装置であるCPUの転送速度が決まってしまうため、上位装置であるCPUからの転送速度をあげて高速に処理することができない。以上のような課題を解決する方法として、第1図のようなディスクドライブの構成で、CPU1

から転送されてきた一つのデータを $n$ 個に分割しデータディスク#1から# $n$ に並列に一度に書き込み、読み出す際は逆にデータディスク#1から# $n$ より一度に読み込む方式が特開平1-250128に開示されている。しかし、このように一括に書き込み、一括に読み出す方式では、大量のデータをまとめて扱う場合は良いが、小さなデータを読み出す場合、データディスク#1から# $n$ を全て使用するためシステムの処理効率が悪い。

〔課題を解決するための手段〕

上記問題を解決するため一括にデータを書き込むときは上位装置であるCPUからシリアルに転送されてきた大量のデータを、データを読み出す際のアクセス単位であるブロックを基準として、マトリックス状のデータ構成に変換し、各当該データ格納用ディスクドライブに対応したマトリックスデータ内のデータをパラレルに書き込む。また、読み出す際は、個々のディスクドライブからブロックを単位として読み出すことで解決できる。

〔作用〕

本発明のようにマトリックスデータを作成してパラレルに書き込み処理を行うことにより個々のディスクドライブの転送速度は同じでもシステムとしての転送速度が上がる。また、磁気ディスクにおける処理時間が短縮されるため、上位装置であるCPUからの転送速度を上げられる。一方読出し時には個々のディスクドライブへのアクセスが可能となるため、処理効率が向上する。

〔実施例〕

実施例1

以下本発明の一実施例を第1図により説明する。本実施例はCPU1、データ制御部(以下DCU)2、データバッファ(以下バッファ)3、データ用磁気ディスク4(以下データディスク)、ECC用磁気ディスク(以下ECCディスク)5、ECCジェネレータ6、Read用データバッファ(以下Readバッファ)7、データ復元部8より構成される。CPU1より発行されたI/O要求はDCU2を通して各磁気ディスクに発行される。

まず、本実施例における具体的なI/O処理に

ついでの大まかな動作を第1図及び第2図を用いて簡単に説明する。本実施例ではデータの記憶又は書き替えはECCを作成するグループ単位(これをECCディスクグループ14という)で一度に大量に行う。CPU1より高速で、シリアルに転送されてきた書き込みデータ9は第1図のDCU2の管理のもとでデータバッファ3内においてECCディスクグループ14に対応したマトリックスデータ10として振り分けられる。なお、書き込みデータ9及びマトリックスデータ10は多数のブロックにより構成されている。このブロックという単位は上位からの1アクセス単位である。つまり、上位(CPU1)から、データを読み込めというRead要求が当該データディスク4へ発行された場合、当該データディスク4からReadする単位は1ブロックである。このように、書き込みデータ9をマトリックスデータ10に展開した後、バッファ3より各ブロックを当該データディスクグループ14内の各データディスク4にまとめて書き込む。また、データディスク4に書き込む際、各々のブ

ロックはECCジェネレータ6にも同時に転送され、各々のブロックからECCを作成しECCディスク5に同様に記録される。読出し時はECCディスクグループ14内における当該データが格納されている当該データディスク4のみにアクセスし当該データを読み出す。本実施例では以上のようにI/Oを処理する。

そこで、以下にI/O処理動作について詳細に説明する。

上記したように本実施例では、一括してデータを書き込むため、シリアルなデータとしてCPU1より転送されてきた書き込みデータ9はマトリックスデータ10に変換する必要がある。そこで、まずこの変換方法について第3図により説明する。本実施例では各データディスク4及びECCディスク5のトラック容量(磁気ディスクの一周に格納できるデータの容量)は40KBであり、各ブロックのデータ量は4KBで、データディスク4の台数は4台である。第3図に示すようにCPU1からの書き込みデータはシリアルにまとめて、大

量に送られてくる。この、まとめて送られてくる書き込みデータ9の量は、常に一定である方が望ましい。この書き込みデータ9の容量はあらかじめDCU2に登録されている。次に、大量の書き込みデータ9は既に前もって登録されている容量(読込み時の1アクセス単位であるブロックの整数倍)に区切られ、n個の領域に分けられる。次に、各領域は、データディスク4の台数分にブロックを単位として区切られる。この4分割されたデータが、それぞれのデータディスク4に格納されるブロックの集まりとなりColumnデータと呼ぶ。以上のようにDCU2の管理のもとでデータバッファ3内において各データディスク4に格納するブロックの集まり(Columnデータ)を、各データディスク4に対応して作成した物をマトリックスデータ10とする。本実施例では書き込む際はECCディスクグループ14内の4台のデータディスク4を1つの論理ボリュームとして考える。つまり、論理的なトラック容量は物理的なトラック容量(個々のデータディスク4及びECCディスク5

のトラック容量)のデータディスク4の台数倍となる。

次に、このようにして作成したマトリックスデータ10を、一括に書き込む方法について第4図を用いて説明する。第4図はデータディスク4及びECCディスク5の内部構成を示す。データディスク4及びECCディスク5はn枚のディスクと、ヘッドを移動させるアクチュエータ11に、n個のヘッドがついて構成される。各々のヘッドは1つのアクチュエータ11についており、アクチュエータ11が移動することにより、全てのヘッドが同じように動く。つまり、アクチュエータ11の動きによりn個のヘッドに対応するn個のディスク上のトラックが決定される。このトラックの集合のことをシリンドラという。磁気ディスクではディスクに同心円状にトラックが設定されているため、外周から内周に向いトラックにアドレスを割り当てる。このアドレスのことをシリンドラ番号という。そこで、R/W回路12は当該ヘッドを選択してトラックを決定しデータの読み寄

きを行なう。各ディスク1周の容量は物理的なトラック容量である。このような構成のデータディスク4、ECCディスク5によりECCディスクグループ14が構成される。まず、領域1を書き込む場合、第3図に示すような領域1に対応したマトリックスデータ10を作成し、各Columnデータを、当該データディスク4に書き込む。第4図に示すように各々の当該データディスク4ではヘッド#1を選択してディスク#1の斜線の部分に第3図における領域1のマトリックスデータ10内の当該データを書き込む。なお、ヘッド#1、2を同時にR/Wできる場合は、同時に行う。ディスク#1に書き込みが終了し領域1のデータの書き込みが終了していない場合、ヘッド#2を電気的な切り替えにより選択し、ディスク#1と同様に、ディスク#2の斜線の部分に書き込む。以下ディスク#3、#4……#nと同様に書き込む。このようにして、領域2、3、4、5……nと同様に書き込んでいく。なお、ディスク#2上のトラックが、ディスク面上の欠陥等により使用できない

このように一括書き込みの際にマトリックスデータ10に対してECCを作成することにより、もしある1台のデータディスク4に障害が発生し、対応したColumnデータが読込み不能になった場合、残りのデータとECCから障害データディスク4に対応したColumnデータを復元することが可能となる。

このように各領域においてマトリックスデータ10を作成し、マトリックスデータ10内のそれぞれのデータディスク4に対応したColumnデータを並列に書き込むことにより高速な処理が可能となる。

以上マトリックスデータ10の作成方法及び格納方法について説明したが、第9図に示すようにCPU1からの書き込みデータ9を領域に区切っていき、最後の領域nにおいてマトリックスデータ10を作成した時、各Columnデータ(各データディスク4に対応して格納されるブロックの集合)の長さが揃わない場合がある。そこで、このような場合の処理について以下に示す。DCU2はマ

場合、現在はあらかじめ用意してある予備のトラックヘッドを移動させ使用している。本発明ではそのような予備のトラックへの切り替えを無視し、ディスク#2をとばしてディスク#3に書き込む。このように、各領域をシリンドラに書き込んでいき、ECCディスクグループ14内のデータディスク4のうち、一つでもディスク#nを書き終えた時点で、そのシリンドラへの書き込みを終了とし、別のシリンドラに移って書き込む。

一方、このように各データディスク4にデータを書き込むと同時に、データをECCジェネレータ6にも転送し、ECCを作成し、ECCディスク5に書き込む。そこで、以下にECC作成方法について説明する。ECC作成方法は、書き込みデータ9をマトリックスデータ10に変換し、第7図に示すように各データディスク4に対応したColumnデータに展開する。そこで、マトリックスデータ10についてパリティ作成グループ13(Rowデータ)においてECCジェネレータ6によりECC(例えば奇数パリティ)を作成する。

トリックスデータ10を作成した時、第9図に示すように、各Columnデータの長さが揃っていないことを認識している。第9図のマトリックスデータ10を見るとデータディスク#4に対応したColumnデータは他のColumnデータと比較し長さが2ブロック分短くなっている。そこで、このようなマトリックスデータ10を各データディスク#4に並列に書き込んでいく場合、データディスク#4に対応したColumnデータはブロック38までの転送を終了した時点で、そのColumnデータの転送を終了とする。この時、すなわち各Columnデータの長さが揃わないことが生じた場合のECC作成方法について説明する。ECCの作成においては、第9図のマトリックスデータ10のデータディスク#4に対応したColumnデータで、ブロック31から38までのRowデータはデータディスク#1から#4に対応したColumnデータを並列にECCジェネレータ6に転送し、それ以降はデータディスク#1から#3のみを並列に転送してECCを作成する。以上のような並列に転送する

データディスク4の台数をダイナミックに切り替えるのはDCU2が制御する。

以上のようにマトリックスデータ10を作成し、データ及びECCを当該ECCディスクグループ14に格納するのだが、その時の書き込むタイミングについて、第16図により説明する。マトリックスデータ10内の各Columnデータを当該ECCグループ14内の各データディスク4に並列にデータを転送し、同時に、ECCジェネレータ6にも同様にデータを転送し、ECCを作成する。しかし、ECCの作成にはある一定の処理時間が必要であり、データディスク3へのデータ格納開始時間とECCディスク5へのデータ格納開始時間との間にずれが生じる。この時、各データディスク4及びECCディスク5への記録タイミングには、次の同期、非同期の2方式が考えられる。同期した場合はデータの格納とECCの格納はセットで考えられ、次の領域の格納は、ECCの格納が終了した時点で開始される。一方、非同期の場合はデータの格納とECCの格納は別と考え、

ECCの格納の終了を待たずに次の領域のデータの格納を開始する。第16図にデータ格納とECC格納を同期して扱った場合と、非同期に扱った場合の書き込み時間についてしめす。この図から、同期した場合、領域1, 2を格納したとき

格納時間=データ書き込み時間+ECC作成のずれ×2  
となる。もしn個の領域まで格納したとすると

格納時間=データ書き込み時間+ECC作成のずれ×n  
となる。一方非同期で領域1, 2を格納した場合

格納時間=データ書き込み時間+ECC作成のずれ  
となり、これはn個の領域を格納した場合も同じである。以上のことからデータの書き込み動作とECCの書き込み動作を非同期に扱う方が書き込みデータ9を一括で書き込む際に必要となる時間が短くなる。以上のことから、データとECCの格納を非同期に扱った方が望ましい。

このようにして、データ及びECCを格納する際、ECCディスクグループ14内の各データディスク4、ECCディスク5の回転制御について次に示す。ECCディスクグループ14内の各デ

ータディスク4、ECCディスク5の回転を同期させることにより、まとめて書き込む場合の平均回転待ち時間(Read/Writeヘッドの下に当該レコードが到達するまでのディスクの回転時間)は、1台の時と同じ1/2回転となる。このことから、各データディスク4、ECCディスク5は回転を同期させて書き込むことが望ましい。

次に、データディスク4内の個々のデータを読み出す場合を第2図を用いて以下に示す。本実施例では、ブロック(読み込み時のアクセス単位)の大きさは4KBである。読み込み時ではECCディスクグループ単位でのアクセスではなく、当該レコードを格納している、当該データディスク4へのアクセスとなる。

CPU1から発行されたRead要求はDCU2においてコマンド解釈され、アドレス変換を行なった後、DCU2の管理のもとで当該データドライブ4に読み込み要求を発行する。当該データドライブ4ではDCU2からのコマンドを受け付け次第CPU1との接続を切り、他のデータディスク4

でのRead処理にDCU2及びバスを空けわたす。当該データディスク4ではDCU2及びバスを空けた後、独自で当該レコードが格納されているトラックまでヘッドを移動させるシーク及び当該レコードがヘッドの下に来るまで回転待ちを行なう。シーク及び回転待ちを行なった後、当該レコードの読み込みが可能になり次第、DCU2およびバスを再び獲得しRead用Data Buffer(RDB)7にデータを転送し、そこからデータを高速でCPU1に転送する。なお、読みだし時はデータディスク4、ECCディスク5の回転を同期しても、しなくても構わない。

次にアドレスの管理方法について説明する。

まず、書き込み時について説明する。書き込み時、CPU1はECCディスクグループ14を論理ボリュームとする。DCU2内にはECCディスクグループと、そのグループ内の個々のデータディスク4、ECCディスク5の使用状況を示す管理テーブルを持っている。このECCディスクグループの管理テーブルと、そのグループ内の個々の

データディスク4、ECCディスク5の管理テーブルは階層的な構造になっている。ECCディスクグループ内のデータディスク4とECCディスク5の中で1台でも使用されている場合、データディスク4及びECCディスク5の使用状況を示す管理テーブル内の該当した使用中であることを示すフラグが“ON”（使用中）になり、同時に所属するECCディスクグループが使用中であることを示すフラグが“ON”（使用中）になる。書き込みに際しCPU1は第19図に示すようにECCディスクグループの使用状況を示す管理テーブルを見て、当該ECCディスクグループ14が使用中であるかフラグを調べる。前のI/O要求により当該ECCディスクグループ14が使用中の場合は再度I/O要求を発行し、使用されていないければ、書き込み処理に入る。このようなフラグを用意しておくことにより、一括書き込みの際当該ECCディスクグループ14内の各々のデータディスク4について使用状況をチェックせずすむ。また、本実施例におけるシステムのトラッ

ク容量は個々のデータディスク4の実際のトラック容量の4倍（ECCグループ内のデータディスク数が4台の場合）とする。本実施例では第2図に示すようにCPU1からはシーケンシャルな書き込みデータ9として送られてくる。この、書き込みデータ9を当該ECCグループ14に格納する場合、その書き込みデータ9はDCU2の管理のもとで、データバッファ3内においてマトリックスデータ10に変換され、ECCディスクグループ14内の各データディスク4に並列同時に書き込まれる。この時同時に以下に示すようにアドレス変換テーブル及び、各データディスク4内のカタログに書き込む。通常、書き込み時はCPU1が書き込みデータ9に対し、データセット名を割り当てて転送する。そこで、DCU2ではCPU1

(OS)が持つデータセット名とDCU2が実際に管理するアドレス（ECCグループナンバ、データディスクナンバ）との対応表として、アドレス変換テーブルをDCU2内部のメモリに格納する。この、アドレス変換テーブルを格納するメモ

リは揮発でも良いが、不揮発が望ましい。さらに、このアドレス変換テーブルはアドレス変換テーブル用ディスク15にも同時に書き込まれ、信頼度を高めることが望ましい。また、書き込みデータ9はマトリックスデータ10状に展開され、個々のColumnデータとして当該データディスク4に格納されるため、それぞれのデータディスク4において格納されるデータのヘッドナンバ、シリンダナンバ、レコードナンバを当該データディスク4内のカタログに記録する。

一方読み込み時ではCPU1(OS)からは当該データセット名を指定し、読み込み可能かを調べる。第19図に示すように、まず、当該ECCディスクグループ14が使用中であることを示すフラグが“OFF”（使用されていない）の場合はそのまま読み込みにかかり、“ON”（使用中）の場合は当該データディスク4が使用中かを調べる。使用可能な場合、第17図に示すようにCPU1より指定されたデータセット名から、DCU2がそのデータセットが格納されているデータディス

ク4のアドレスを割り当てる。この割り当てる方法について以下に説明する。第18図に示すようにCPU1から指定されたデータセット名から当該ECCディスクグループ14を見付ける。第17図に示すように、そのアドレス変換テーブルにおいて当該データセットが格納されているアドレスがaからbまでならデータディスク#1に対応し、bからcならデータディスク#2、cからdならデータディスク#3、dからeならデータディスク#4に対応する。このように、当該データセットが格納されている、当該データディスク4のアドレスを見つけ、次に、第5図に示すように、そのデータディスク4内のカタログにより当該データが格納されているヘッドナンバ、シリンダナンバ、レコードナンバを調べ、当該データを読み出す。なお、上記アドレス管理方法においては第5図に示すようにカタログを各データディスク4内に置いたが、高速なアクセスを要求する場合は、このカタログをDCU2内のアドレス変換テーブルに置く方が望ましい。このようにするこ



とにより、CPU1から転送されてきたデータセット名に対し、即座にデータディスクナンバ、ヘッドナンバ、シリンダナンバ、レコードナンバを割当て、高速に当該データをアクセスすることが可能となる。また、各データディスク4内のみにカタログを置くのではなく、各データディスク4にカタログ用のバッファを設け、いちいちカタログを読み込みデータディスク4をアクセスせずに、高速に当該データのアドレスを見つけることも可能となる。

以上述べてきたように、本実施例を用いることにより、書き込みデータ9をマトリックスデータ10に変換する時の変換アルゴリズムは簡単になる。また、個々のブロックの管理も楽である。

本実施例では、トラック容量、ブロックのデータ量、データディスクの台数を変えた場合でも、同様な効果が期待できることは明らかである。また、データディスク4及びECCディスク5を磁気ディスクとしたが、光ディスク、フロッピーディスクなどを使用しても同様なことが可

能である。第6、8図に示すように、不足しているデータ分だけ“0”あるいは任意のデータを書き込んでおく。このようにして、見かけ上各Columnデータの長さが等しいマトリックスデータ10を作成し、通常のECC作成アルゴリズムによりECCを作成する。以上のように、DCU2によりマトリックスデータ10の各Columnの長さがたりない部分にALL0のデータを書き込み、長さを揃えた方が、ECC作成アルゴリズム内に、長さが揃わない場合のアルゴリズムを用意する必要がないため、単純化されて良い。

なお、本実施例では長さがたりない分、“0”を書き込むことにしたが、“0”以外の特定のパターンを書き込んでも良い。

また、本実施例は実施例1に記載の他のすべての動作に適用可能である。

### 実施例3

以下にマトリックスデータ作成方法についての他の実施例を実施例3として示す。本実施例ではまとめて一括書き込みを行う際、書き込みデータ9を、

能である。

### 実施例2

DCU2がマトリックスデータ10を作成した時、各Columnデータの長さが揃っていない場合の処理方法の他の実施例を実施例2として以下に示す。第9図でCPU1からの書き込みデータ9を領域に区切っていき、最後の領域nにおいてマトリックスデータ10を作成した時、各Columnデータ（各データディスク4に対応して格納されるブロックの集合）の長さが揃わない場合がある。本実施例でも実施例1と同様に一括書き込みの際に第7図に示すようにECCを作成し、障害発生時は残りのデータとECCより障害データディスク4に対応したColumnデータを復元することが可能となる。そこで、このような場合のECC作成方法について次に説明する。DCU2はマトリックスデータ10を作成した時、第9図に示すように、各Columnデータの長さが揃っていないことを認識している。

そこで、本実施例では、Columnデータの長さを

Row方向にデータを作っていく（当該ECCディスクグループ14の各データディスク4に順番に振り分けていく）マトリックスデータ10に変換する方法である。本実施例では実施例1と同様、各データディスク4及びECCディスク5のトラック容量（磁気ディスクの一周の容量）を40KB、各ブロックのデータ量を4KBとしデータディスク4の台数を4台とする。

第10図に示すようにCPU1よりまとめて大量に転送されてきた書き込みデータ9はDCU2の管理のもとでマトリックスデータ10に変換される。まず、第11図に示すように、実施例1と同様に書き込みデータ9はn個の領域に分けられる。次に、各領域は、データディスク4の台数分に振り分けられる。この振り分けられたデータが、それぞれのデータディスク4に格納されるブロックの集まりとなる。以下に振り分ける方法について第12図を用いて具体的に説明する。CPU1より転送されてきた書き込みデータ9は多数のブロックにより構成され、シリアルに転送されてくる。

この書き込みデータ9内で1個以上のブロックの集合（この集合の大きさは等しくする）を作り、書き込むデータディスク4に対応したColumnデータとして振り分けていく。書き込みデータ9でBlock 1からBlock Lまでを第1回のデータディスク#1に格納すべくColumnデータ#1に振り分け、Block L+1からBlock mまでをデータディスク#2に格納すべくColumnデータ#2に振り分ける。同様にColumnデータ#3、#4にBlock m+1からn、n+1からpを振り分ける。本実施例ではデータディスクの台数を4台としたため、Columnデータ#4に割当てた後は再びColumnデータ#1にBlock p+1からqを割当て以下同様に書き込みデータ9をColumnデータに展開していく。以上のようにして、書き込みデータ9をマトリックスデータ10に変換する。

このようにマトリックスデータ10を作成することにより、一括書き込みで並列にしかも高速に書き込むことが可能となる。

次に、この変換方法を使ったとき、マトリックス

データ10のデータディスク#3、4に対応したColumnデータで、ブロック3から35、4から36までのRowデータはデータディスク#1から#4に対応したColumnデータを並列にECCジェネレータ6に転送し、それ以降はデータディスク#1と#2のみを並列に転送してECCを作成する。以上のような並列に転送するデータディスク4の台数をダイナミックに切り替えるのはDCU2が行うものとする。

次に、実施例2と同様にColumnデータの長さを揃える目的で、第14図に示すように、不足しているデータ分だけ“0”あるいは任意のデータを書き込んでおく方法について説明する。以上のように、DCU2によりマトリックスデータ10の各Columnの長さがたりない部分にALL“0”あるいは任意のデータを書き込み、見かけ上各Columnデータの長さが等しいマトリックスデータ10を作成し、通常のECC作成アルゴリズムによりECCを作成する。このように長さを揃えることで、ECC作成アルゴリズム内に、長さが揃

データ10内において各Columnデータの長さが揃わない場合の処理方法について述べる。第13図のように書き込みデータ9からマトリックスデータ10を作成した時、各Columnデータの長さが揃わない場合が生じる。このような場合のデータ書き込み、ECC作成処理方法は以下のようにすれば良い。本実施例では実施例1と同様に、一括書き込みの際に第7図に示すようにECCを作成する。DCU2はマトリックスデータ10の作成時に、第9図に示すように、各Columnデータの長さが揃っていないことをDCU2は認識している。第13図のマトリックスデータ10を見るとデータディスク#3、4に対応したColumnデータは他のColumnデータと比較し長さが1ブロック分短くなっている。そこで、このようなマトリックスデータ10を各データディスク4に並列に書き込んでいく場合、データディスク#3、4に対応したColumnデータはブロック35、36までの転送を終了した時点で、そのColumnデータの転送を終了とする。ECCの作成においては、第13図のマ

トリックスデータ10のデータディスク#3、4に対応したColumnデータで、ブロック3から35、4から36までのRowデータはデータディスク#1から#4に対応したColumnデータを並列にECCジェネレータ6に転送し、それ以降はデータディスク#1と#2のみを並列に転送してECCを作成する。以上のような並列に転送するデータディスク4の台数をダイナミックに切り替えるのはDCU2が行うものとする。

次に、本発明を適用しない場合と実施例1の書き込みデータ9を一端データバッファ3に格納した後、4等分して当該ECCディスクグループ14の各データディスク4に書き込む方法と、書き込みデータ9をデータバッファ3に書き込みながら当該ECCディスクグループ14の各データディスク4に書き込む方法において、各々の書き込み時間の比較を第15図を用いて説明する。第15図に示したようにデータディスク4の転送レートを3MB/s、回転数を3600r.p.m、トラック容量を40KB/TRKとし、ECCディスクグループ14内にはデータディスク4を4台持っているものとする。ECCディスクグループへの転送データ量は

$$40\text{KB/TRK} \times 4\text{台} = 160\text{MB/Group}$$

とした場合、このデータを格納する場合本発明を適用しないと4回転ディスクが回転しなければならない。この時の書き込み時間は

ディスク1回転にかかる時間(16.6ms)×4回転=66.4msとなる。実施例1の方法ではデータバッファ3への書き込み転送レートが12MB/sではトータルの書き込み時間(160MB)は30msとなり、本発明を適用しない場合と比較し45%に減少する。また、データバッファ3への書き込み転送レートを24、48MB/sとしていくと、トータルの書き込み時間はさらに、23.3、20msと減少する。一方、本実施例のようにデータバッファ3に書き込みながら、当該ECCディスクグループ14の各データディスク4に書き込む場合は、データバッファ3への書き込み転送レートを12MB/s以上にすると、データバッファ3への書き込み時間は、データディスク4への書き込み時間中に行われるので全書き込み時間は、データディスク4への書き込み時間となり16.6msとなり本発明を適用しない場合と比較し25%に減少する。以上のことから本発明を適用することにより書き込み時間は減少し、また、データバッファ3への書き込み転送レートが低い場合は本実施例が有効で

ある。

なお、トラック容量、ブロックのデータ量、データディスクの台数を変えた場合でも、本実施例で示すようなマトリックスデータ作成方法を同様に使用できることは明らかである。また、本実施例は実施例1、2に記載の他のすべての動作に適応可能である。

#### 〔発明の効果〕

以上説明したように、一括書き込み、個別読み出しするような、データベースなどでは、本発明のように書き込みをまとめて並列に高速で行うことにより、書き込み時間を短縮することができる。さらに、一括書き込み時に領域に分割しマトリックス状のデータに変換して並列に書き込む。その際、各データ書き込み用磁気ディスクに対応した、データの長さが揃わない場合、その差を詰めないで使用することにより、高速に処理することが可能となる。

#### 4. 図面の簡単な説明

第1図は本発明の全体構成ブロック図、第2図

は第1の実施例のデータ処理動作説明のためのブロック図、第3図は本発明の第1の実施例のデータ変換概念図、第4図は本発明のデータディスク、ECCディスクの内部構成図、第5図は本発明のアドレス変換の概念図、第6図は第1の実施例のデータ変換概念図、第7図は本発明のECC作成概念図、第8図は第2の実施例のECC作成の説明のための概念図、第9図は第1の実施例のデータ変換の説明のための概念図、第10図は第3の実施例のデータ処理動作説明のためのブロック図、第11図は第3の実施例のデータ変換説明のための概念図、第12図は第3の実施例のデータ変換詳細を示す概念図、第13図は第3の実施例のデータ変換を説明する概念図、第14図は第3の実施例のデータ変換を説明する概念図、第15図は本発明の書き込み処理時間比較図、第16図は本発明のデータ書き込み時間の比較タイムチャート、第17図は本発明のアドレス変換方法を説明するブロック図、第18図は本発明のアドレス決定方法を説明するブロック図、第19図は本発明の管理

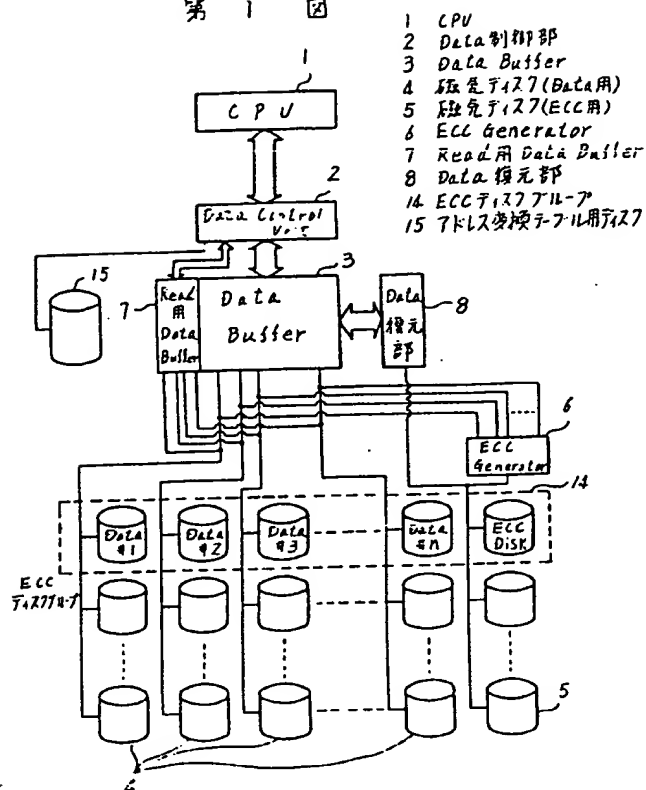
テーブルへのアクセスフローチャート。

1…CPU(中央処理装置)、2…DCC(Data制御部)、3…データバッファ、4…データディスク(データ用磁気ディスク)、5…ECCディスク(ECC用磁気ディスク)、6…ECCジェネレータ、7…Read用データバッファ、8…データ復元部、9…書き込みデータ、10…マトリックスデータ、11…アクチュエータ、12…R/W回路、13…パリティ作成グループ、14…ECCディスクグループ、15…アドレス変換テーブル用ディスク。

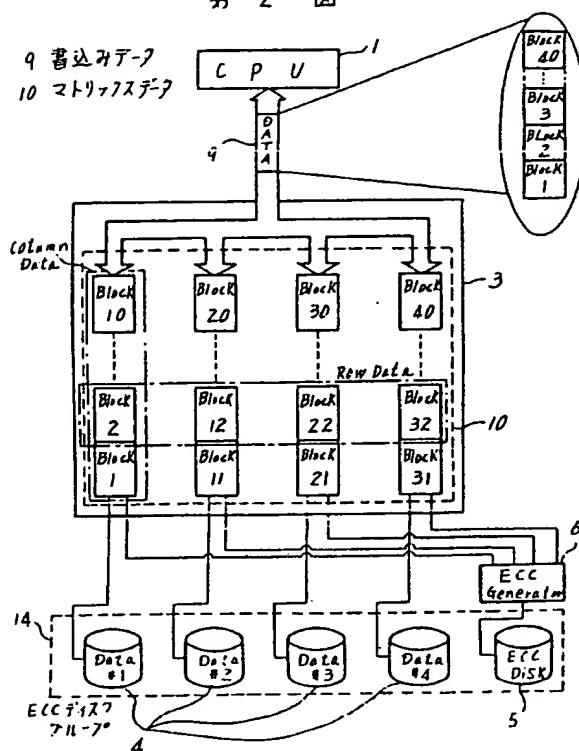
代理人 弁理士 小川勝男



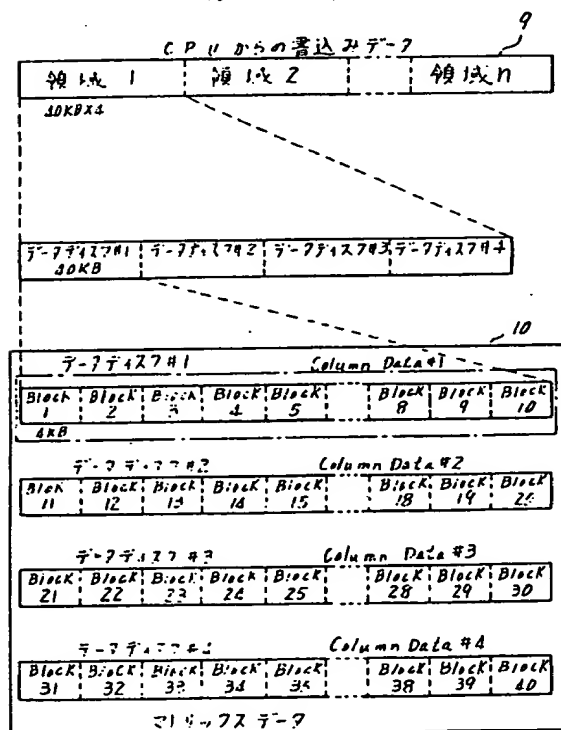
第 1 圖



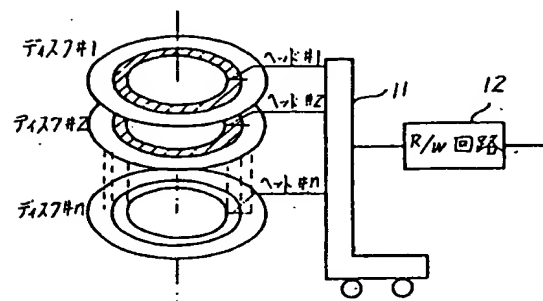
第 2 図



第 3 図



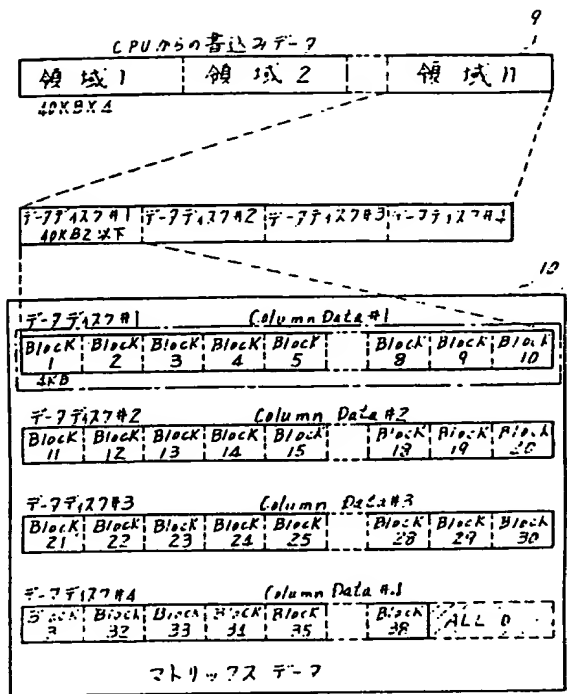
第 4 回



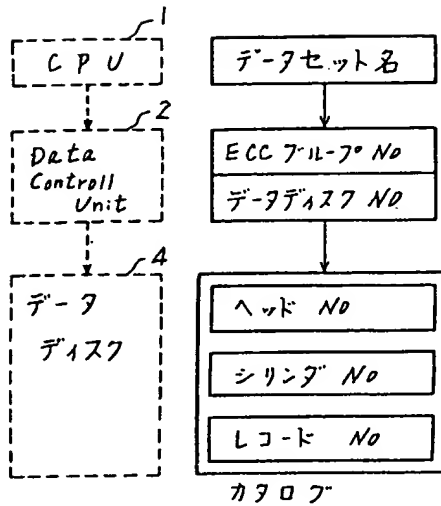
11 37421-7

12 R/w回路

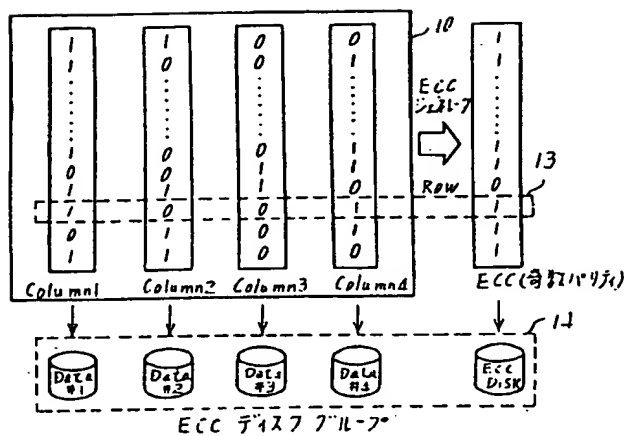
第 6 図



第 5 図

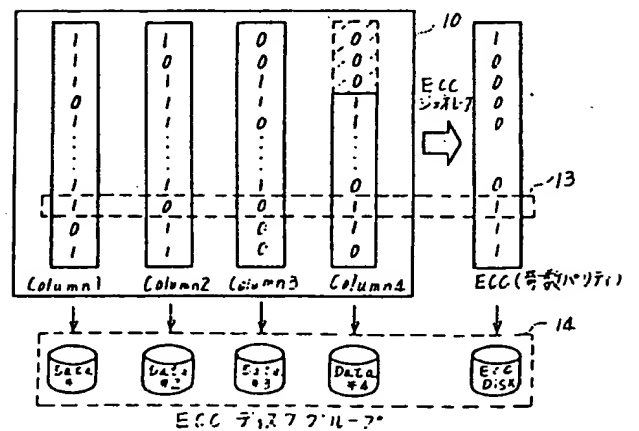


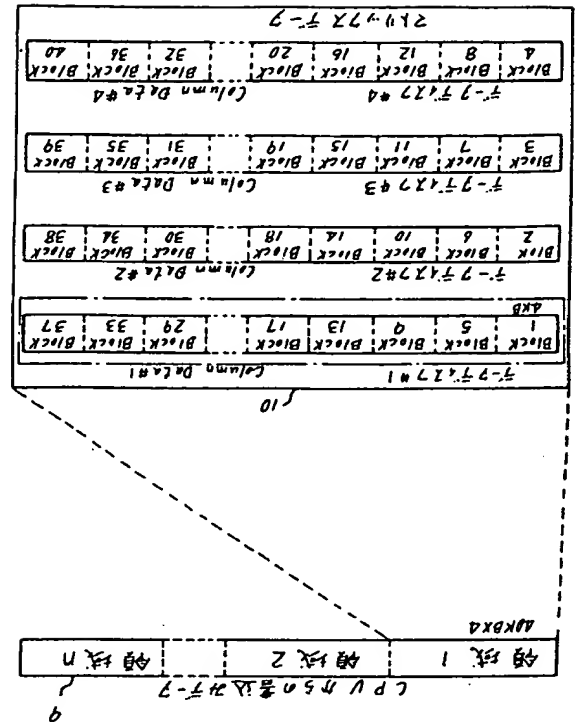
第 7 図



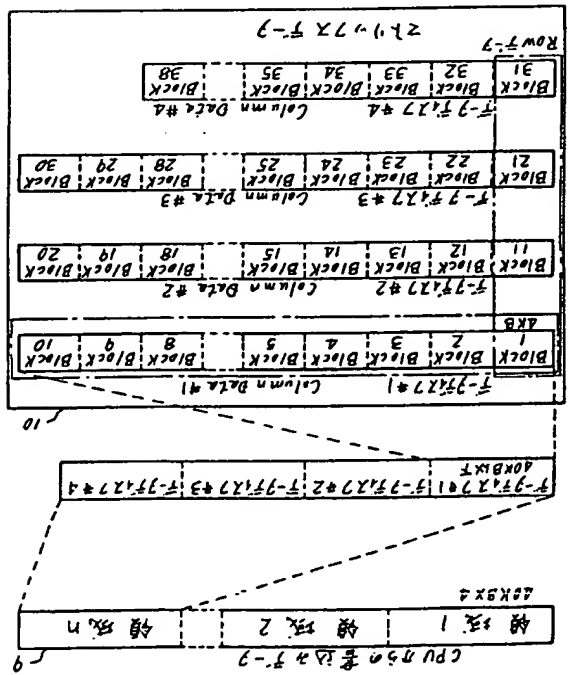
13 パリティ作成  
フィールド

第 8 図

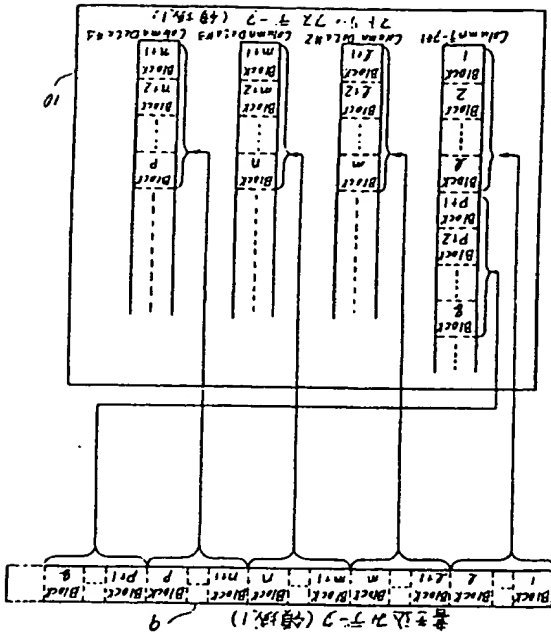




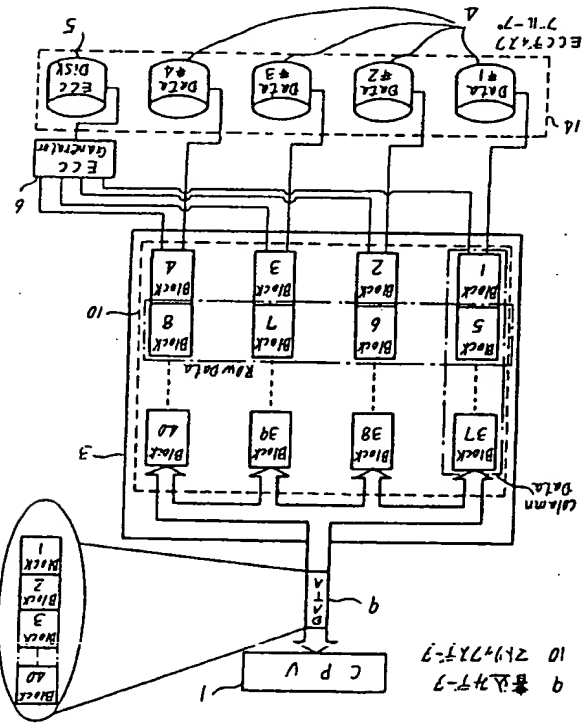
第 11 図



第 9 図

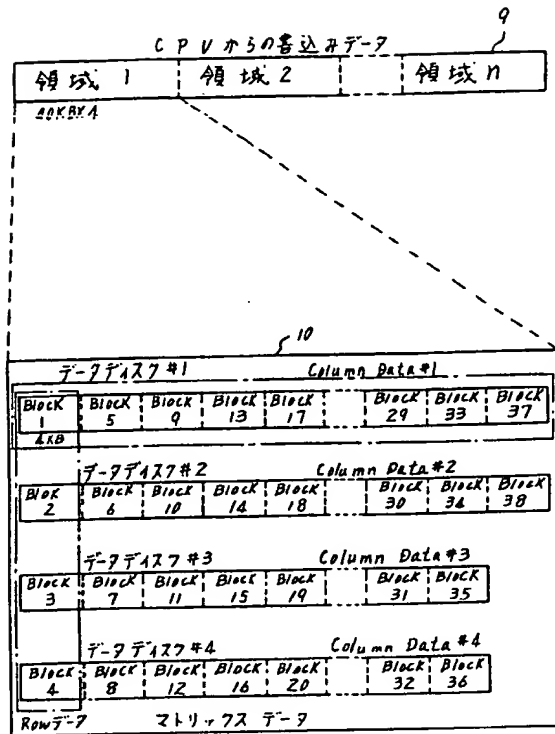


第 12 図

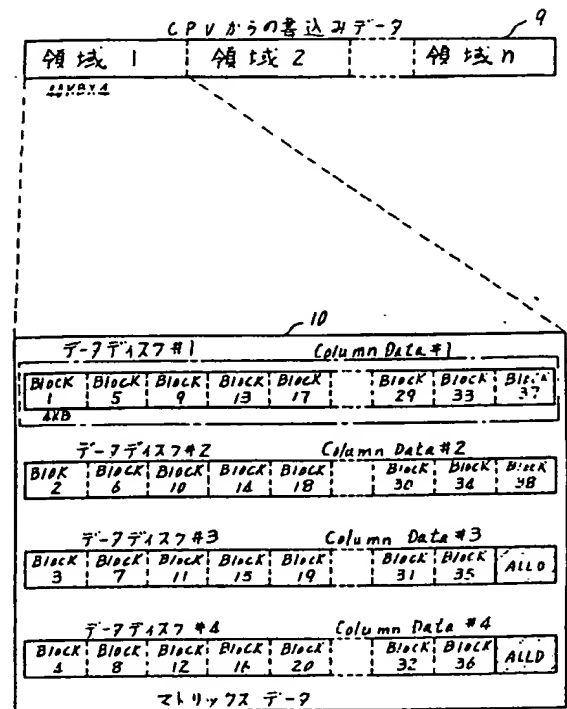


第 10 図

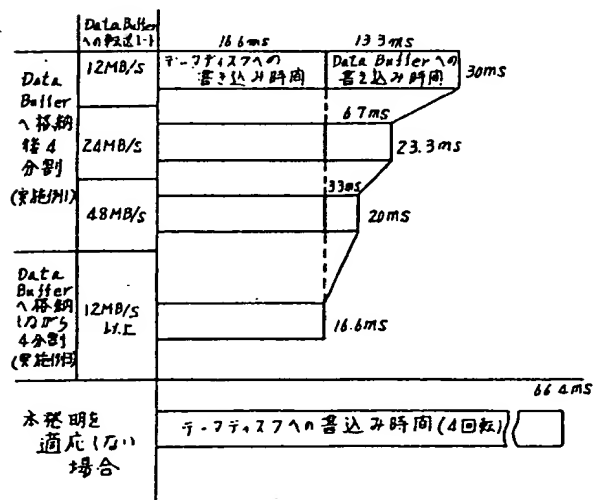
第 13 図



第 14 図

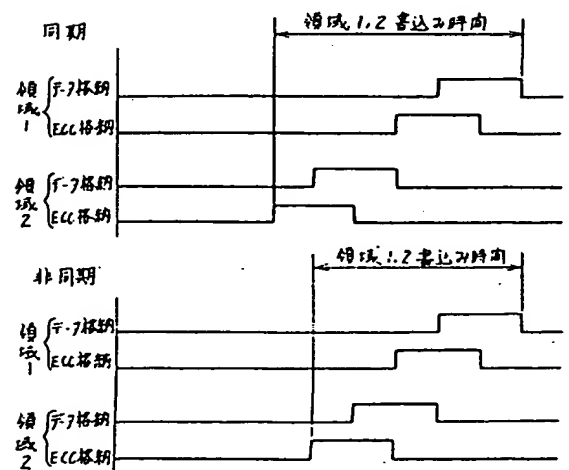


第 15 図

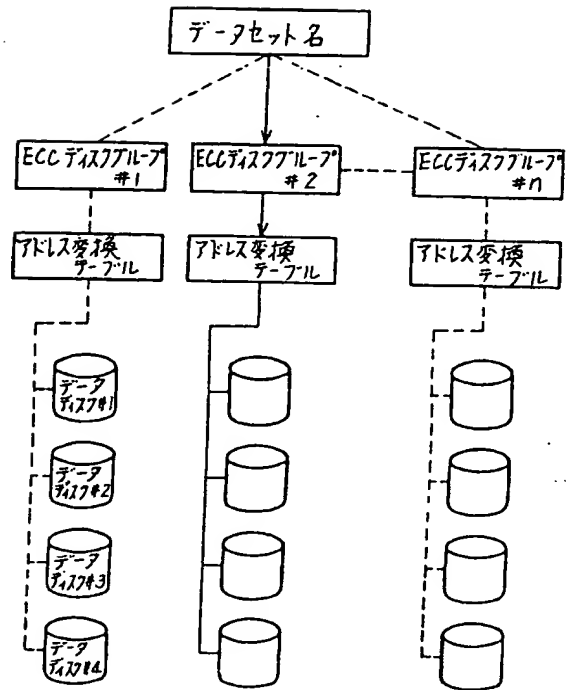
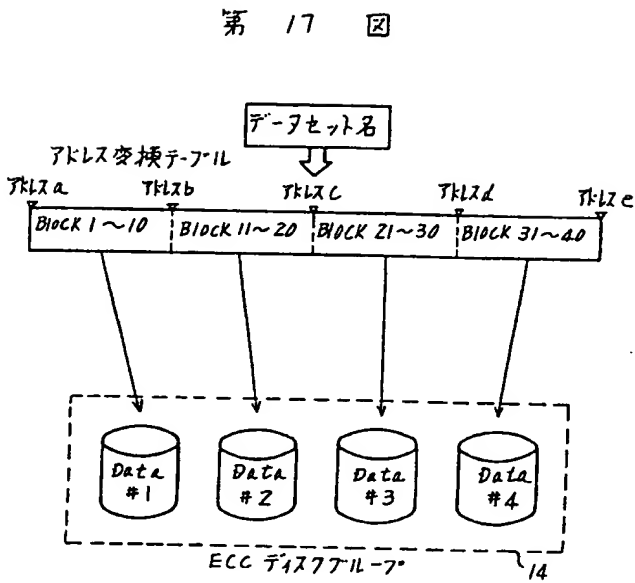


データディスクの転送レート = 3MB/s  
 回転数 = 3600 RPM  
 トラック容量 = 40 KB/TRK  
 ECC71L-7の内蔵データ量 = 40 KBx40  
 = 160 KB/group

第 16 図



第 18 図



第 19 図

